TOPOLOGICAL SORTING

GROUP MEMBERS

A.SHEREEN FATHIMA(140071601068)

D.STELLA(140071601073)

CSE “B”

ALGORITHM DESIGN AND ANAYLSIS LAB(CSB3105)

B.S ABDUR RAHMAN UNIVERSITY

PREFACE

Problem Description

Student Activities

Topological sort

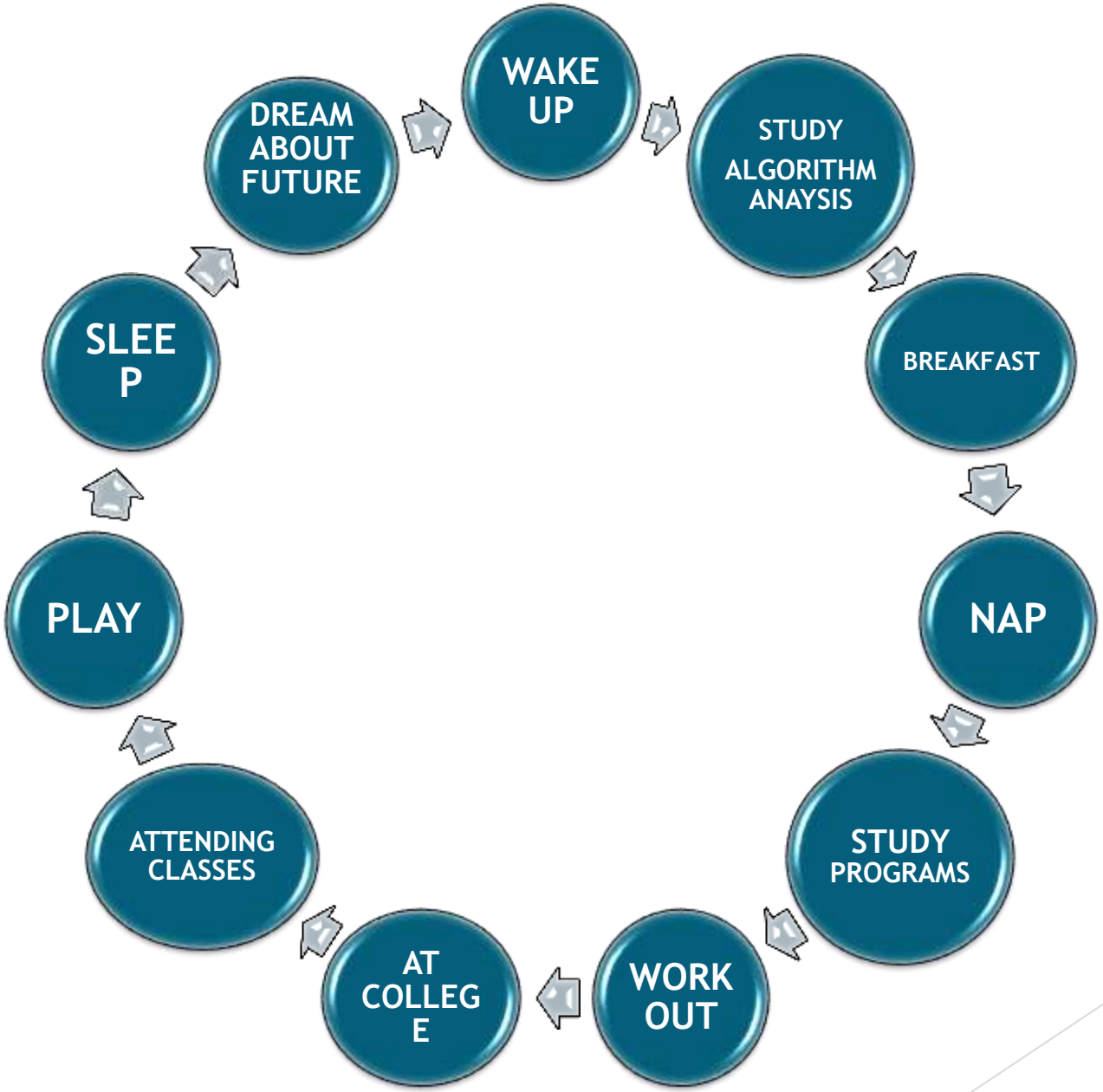
Algorithm of topology sorting

Complexity Of an algorithm

PROBLEM DESCRIPTION

A Student has various activities to be performed on time to maintain effectiveness in both educational and professional development. So in order to keep the activities organized in a effective manner, the order of the activities to be done is sorted systematically using the topological sorting algorithms.

ACTIVITIES OF STUDENT



Student activity graph

TOPOLOGICAL SORT

DEFINITION :

A Topology sort or topological ordering of a directed graph is a linear ordering of its vertices such that for every directed edge UV from vertex U to vertex V,Ucomes before V in the ordering

ALGORITHM

1. Compute the indegrees of all vertices

2. Find a vertex U with indegree 0 and print it (store it in the ordering)If there is no such vertex then there is a cycle and the vertices cannot be ordered. Stop.

3. Remove U and all its edges (U,V) from the graph.

4. Update the indegrees of the remaining vertices.

5. Repeat steps 2 through 4 while there are vertices to be processed

IMPROVED ALGORITHM

After the initial scanning to find a vertex of degree 0, we need to scan only those vertices whose updated indegrees have become equal to zero.

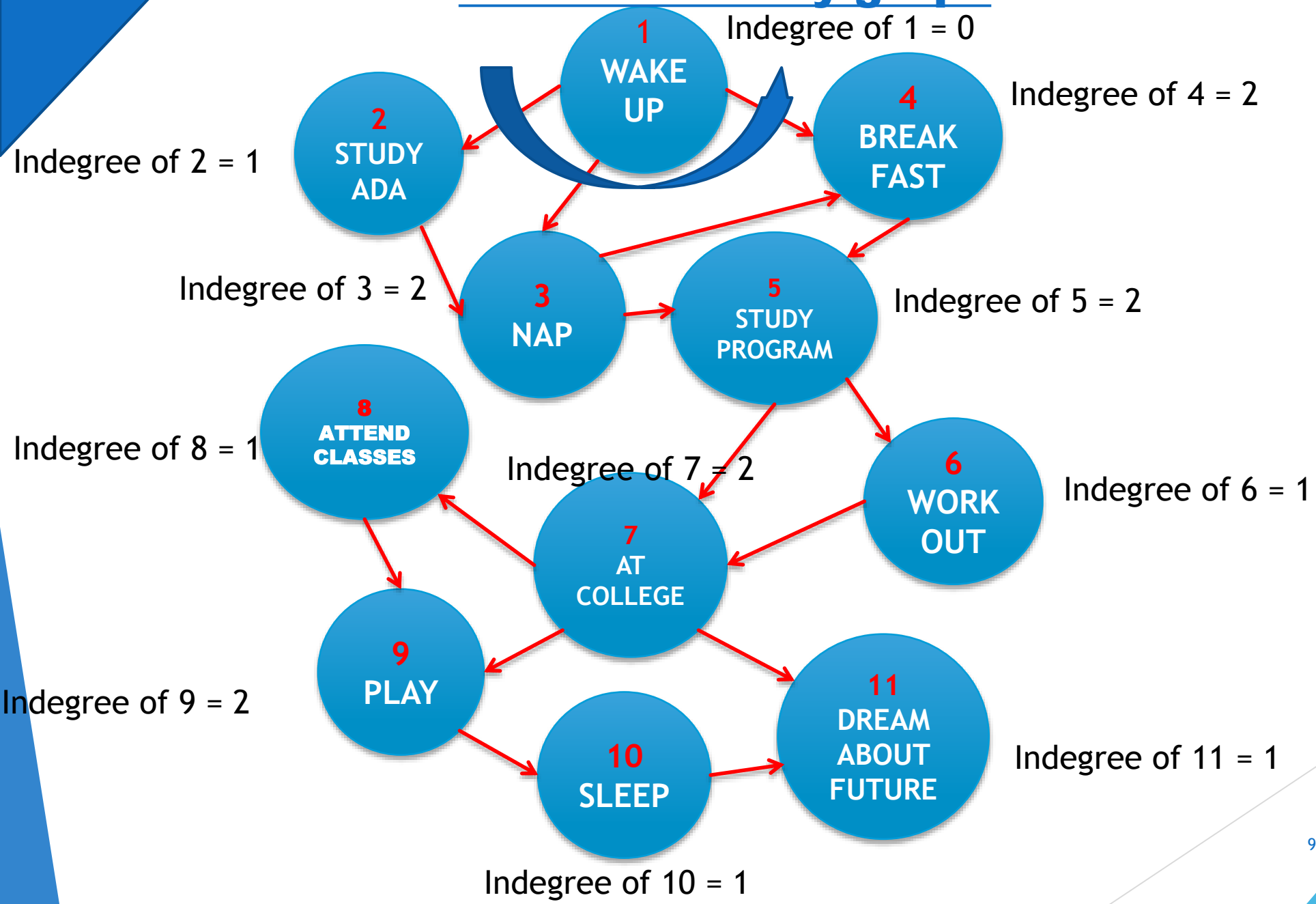
1. Store all vertices with indegree 0 in a queue

2. Get a vertex U and place it in the sorted sequence (array or another queue).

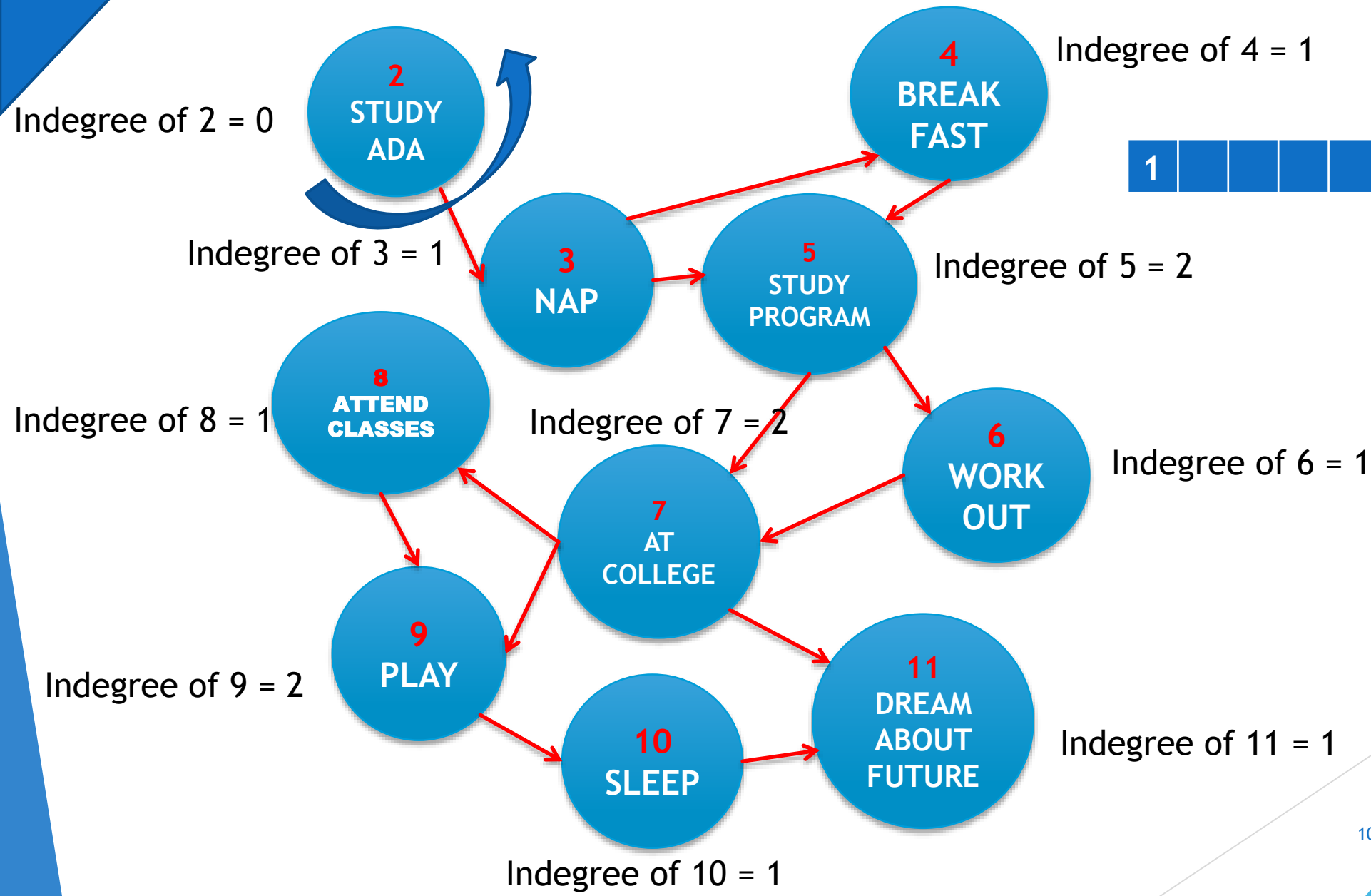
3. For all edges (U,V) update the indegree of V, and put V in the queue if the updated indegree is 0.

4. Perform steps 2 and 3 while the queue is not empty.

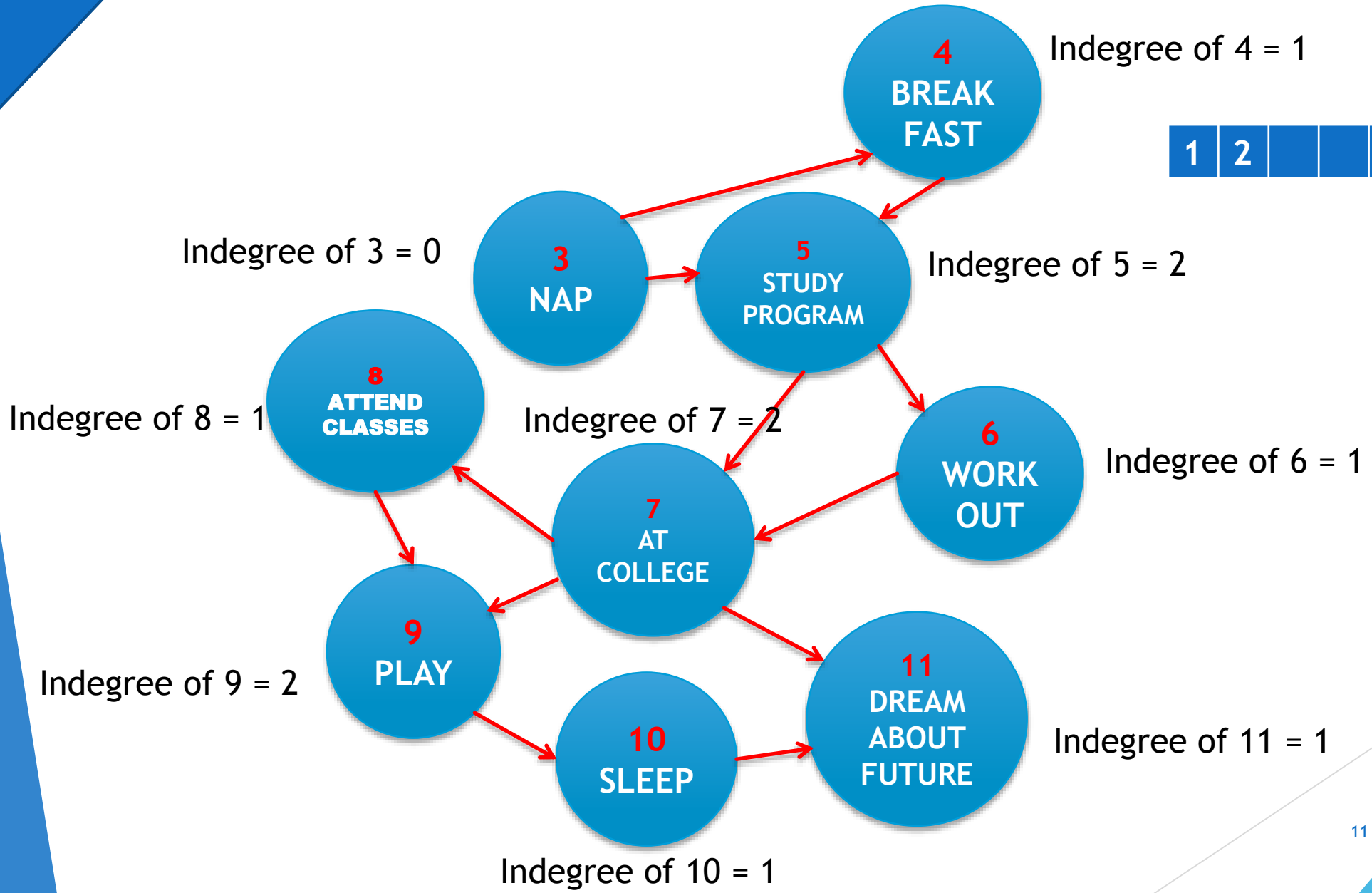
Student activity graph



Student activity graph



Student activity graph



SORTED ACTIVITIES

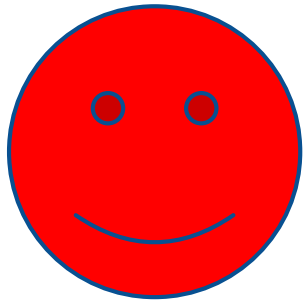
1 WAKE UP	2 STUDY ADA	3 NAP	4 BREAK FAST	5 STUDY PROGRAM	6 WORK OUT	7 AT COLLEGE	8 ATTENDING CLASS	9 PLAY	10 SLEEP	11 DREAM ABOUT FUTURE

COMPLEXITY OF AN ALGORITHM

For input graph G = (*V,E), Run Time = ?*

Break down into total time required to:

- **Initialize In-Degree array: O(|*E|)***
- **Find vertex with in-degree 0:|*V| vertices, each takes O(|V|) to search In-Degree array.***
- **Total time = O(|*V|²)***
- **Reduce In-Degree of all vertices adjacent to a vertex: O(|*E|)***
- **Output and mark vertex: O(|*V|)***
- **Total time= O(|V|² + |E|) Quadratic time!**



THANK YOU